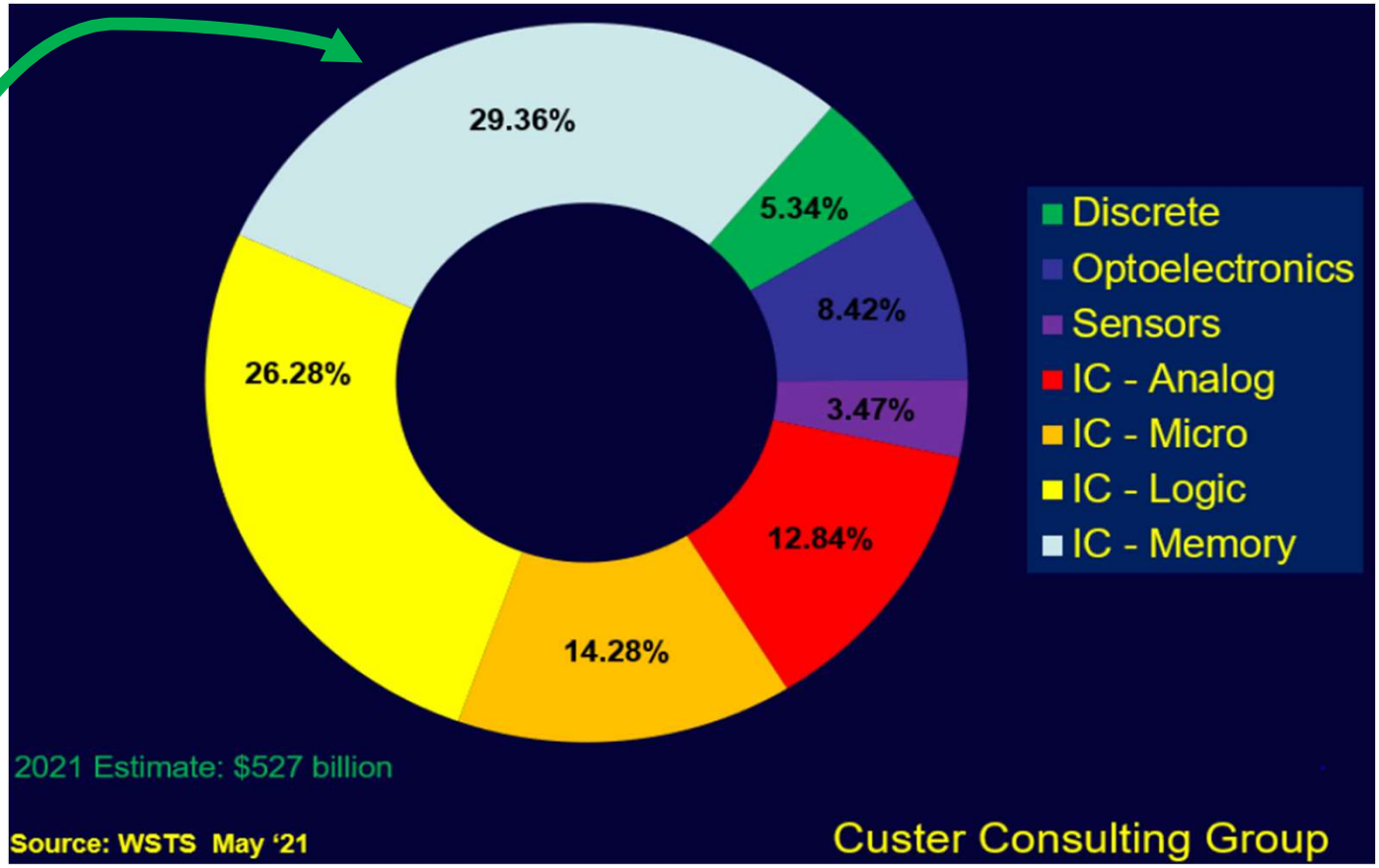
Lezione 17
Introduzione al sottosistema di memoria

Patterson: 5.1,5.2

Market Share dei Principali Componenti a Semiconduttore



Le memorie rappresentano il 29% del mercato dei semiconduttori

Tecnologie usate per realizzare le memorie

- Tecnologie ad accesso sequenziale
  - Il tempo di accesso dipende in maniera lineare dalla locazione fisica
    - Nastri

es. Nastro

- Tecnologie ad accesso semicasuale

~> dischi

  - Il tempo di accesso dipende sia dalla locazione fisica dell'informazione che dal particolare istante in cui faccio accesso
    - Dischi: dischi-fissi, CDROM, DVD, ...

sposto una testina in un punto facendo girare un disco

se sono fortunato la trovo subito, altrimenti aspetto quasi un giro

- Tecnologie ad accesso casuale (Random Access)
  - Il tempo di accesso è indipendente dalla locazione fisica dell'informazione nel dispositivo fisico
    - Memoria principale
    - Due tipi: Non-Volatili e Volatili

Memorie accesso casuale di tipo Non-Volatile (NVM)

- «Non-Volatile» → conservano i dati anche se tolgo l'alimentazione
- ROM (Read-Only Memory)
  - Non scrivibile (sia vantaggio che svantaggio)
- EPROM (Erasable Programmable Read-Only Memory)
  - ✓ Scrivibile (ma una sola volta)
  - ✓ Cancellabile a livello di chip (raggi UV)
  - ☹ Necessario intervenire con apposito "kit di cancellazione"

[1]

- EEPROM (Electrically-Erasable Programmable Read-Only Memory)
  - ✓ Scrivibile (1 sola volta)
  - ✓ Cancellabile a livello di byte (elettricamente)
  - ☹ Dovendo scrivere molti byte si perde molto tempo nelle cancellazioni
- Memoria "Flash": Evoluzione della EEPROM
  - ✓ Scrivibile (1 sola volta)
  - ✓ Cancellabile a livello di blocco (elettricamente)

→ Per rendere possibile «l'illusione di leggere/scrivere» occorre aggiungere un apposito controller che effettua in modo opportuno scritture e cancellazioni

[1] https://en.wikipedia.org/wiki/EPROM#/media/File:NEC_D8749HD.png

Memorie accesso casuale di tipo Volatile

- «Volatile» → conservano i dati solo se alimentate
- Principali tipi di implementazioni:
  - DRAM (Dynamic Random Access Memory)
    - ✓ Dense (molti bit per unità di superficie), basso consumo, basso costo
    - ☹ Lente, dinamiche → Necessità di essere "rinfrescate" periodicamente
  - SRAM (Static Random Access Memory)
    - ✓ Veloci, statiche → basta alimentare il chip per non perdere i dati
    - ☹ Bassa densita', alto consumo, alto costo
- Nuove implementazioni di DRAM:
  - DDRx, HBM, …

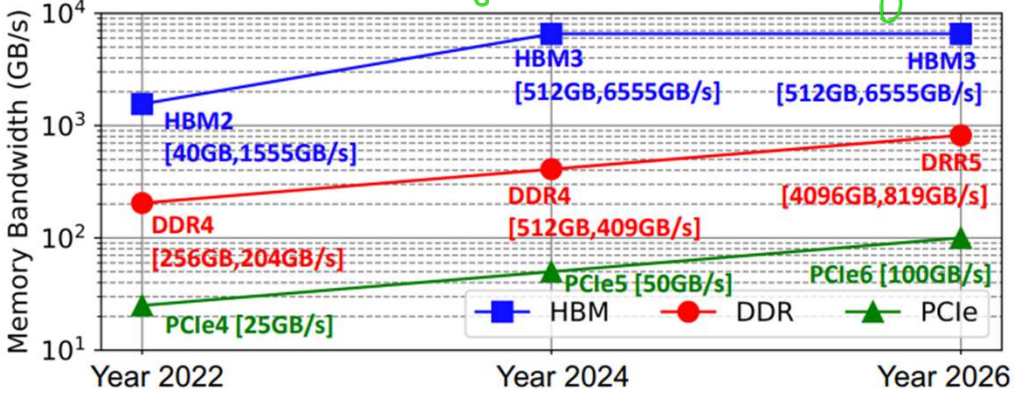


Fig. 1: Memory bandwidth trends for HBM, DDR and PCIe from years 2022 to 2026. Relative bandwidth improvements remain constant. The PCIe is the performance bottleneck for disaggregated systems.

Joël Lindegger and others, Scrooge: a fast and memory-frugal genomic sequence aligner for CPUs, GPUs, and ASICs, Bioinformatics, Volume 39, Issue 5, May 2023

Parametri delle Prestazioni della Memoria

Grandezze da tenere d'occhio per la memoria

t_a=tempo di accesso (Access Time)

inizio dell'oper. di lettura e arriva il primo dato

- Intervallo fra l'inizio di una lettura (indirizzo su A-bus) e l'arrivo del PRIMO dato (su D-bus) $t_a = T_{INIZIO\ LETTURA} \rightarrow T_{RICEVUTO\ PRIMO\ DATO}$

t_c=tempo di ciclo (Cycle Time)

Tra oper. di lettura e una successiva comprende tempi stabilizz. di segnali

- Intervallo fra l'inizio di una lettura e l'inizio della lettura successiva

t_l=tempo di latenza (Latency Time)

t in cui arriva tutto il dato

- Tempo di accesso al DATO COMPLETO ovvero alla k-esima word nel caso in cui la memoria supporti trasferimenti a gruppi di k word (se k=1 t_l=t_a). Es. dischi, ma anche memorie RAM

se ho dato esteso in più byte

inizio di oper. – fine lettura completa

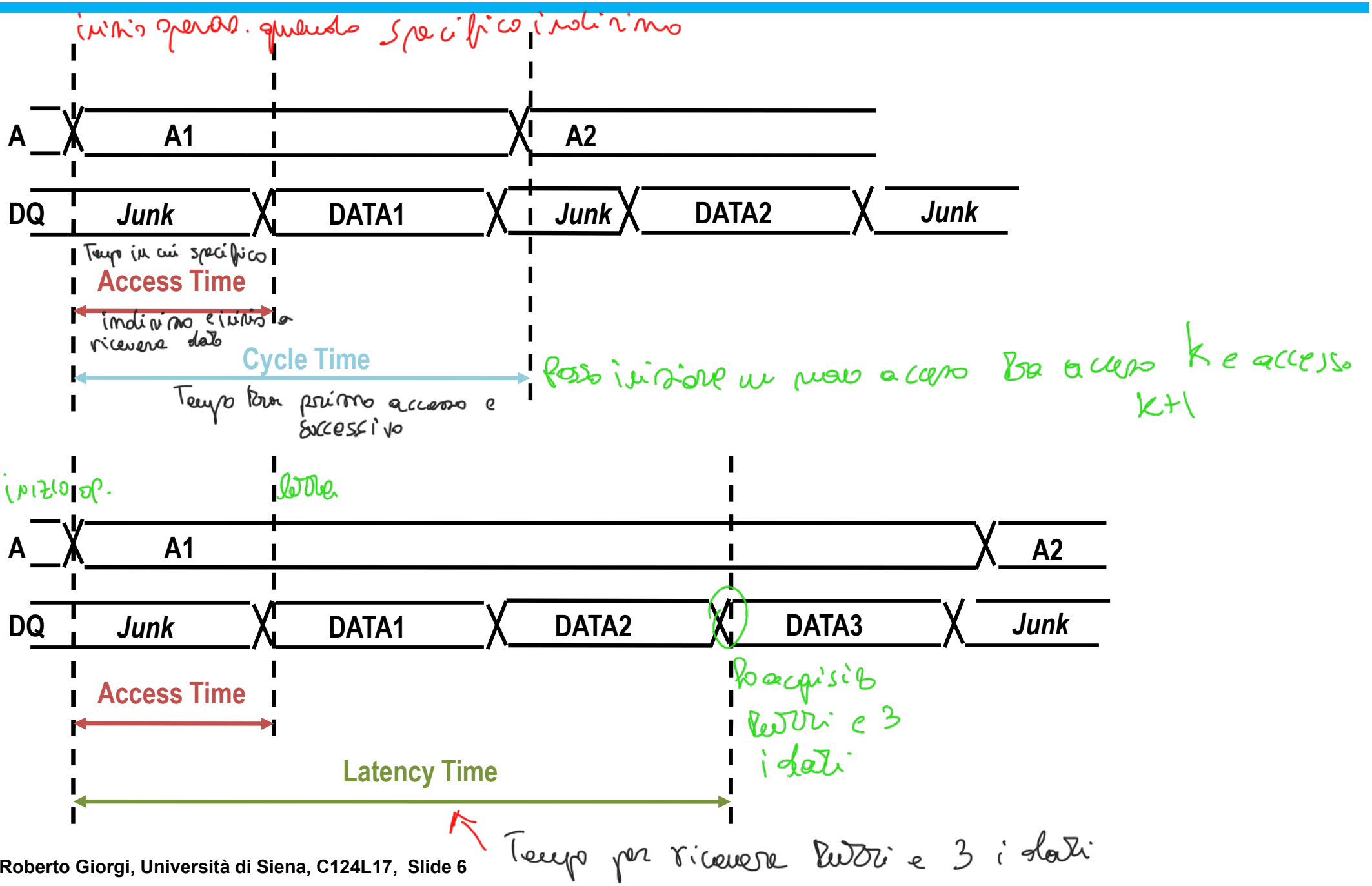
ω=Banda (Bandwidth)

- Tasso di trasmissione dei byte (si misura in byte/s)

Byte/s che riesco a trasmettere

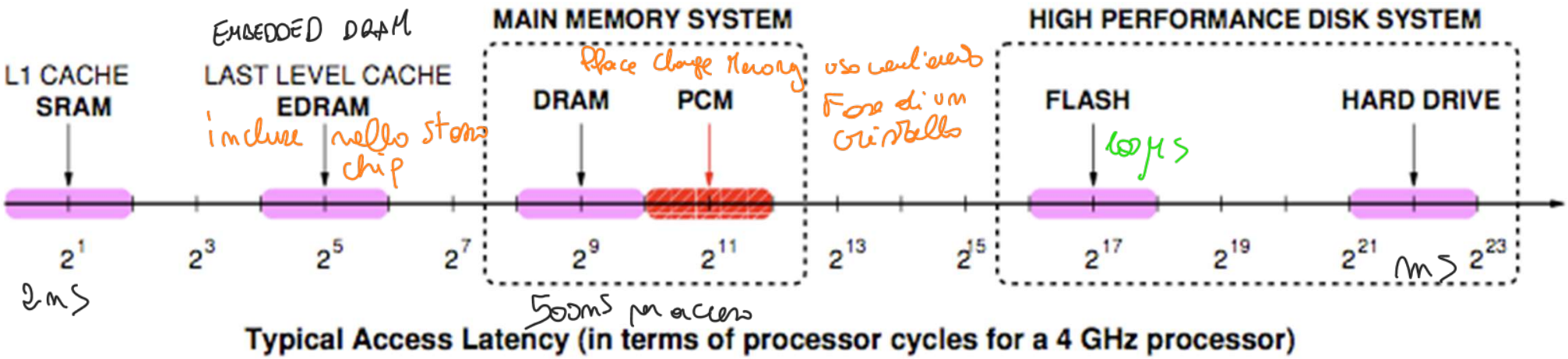
- A parità di tempo di ciclo posso trasmettere più byte in parallelo

Tempi caratteristici delle memorie



DRAM versus SRAM

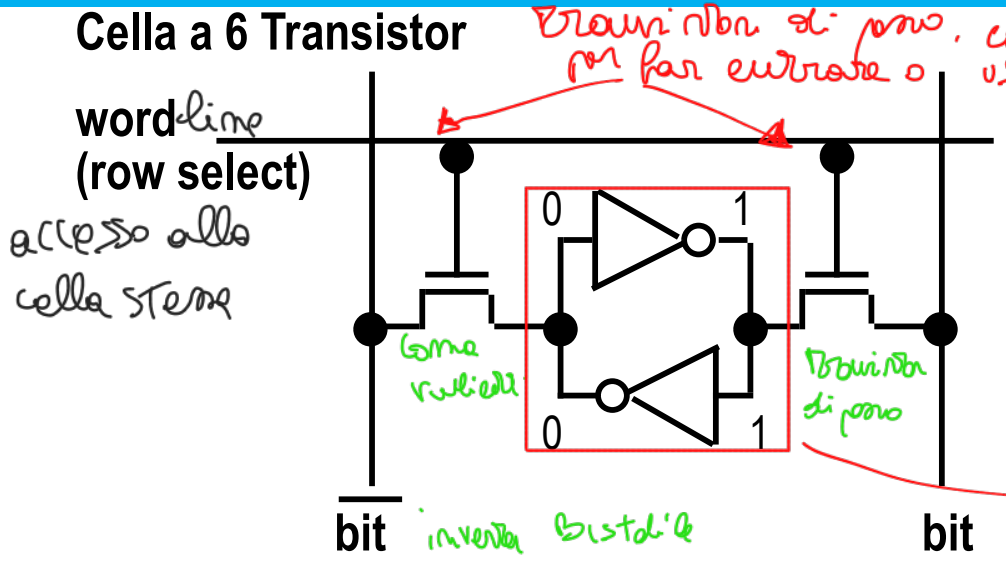
- **Per massimizzare il parametro prestazioni/costo**
  - **Si usa poca memoria SRAM (costosa e veloce) all'interno del chip (es. registri)**
  - **Si usa una ragionevole quantità di DRAM (meno costosa e relativamente veloce) all'esterno del chip**

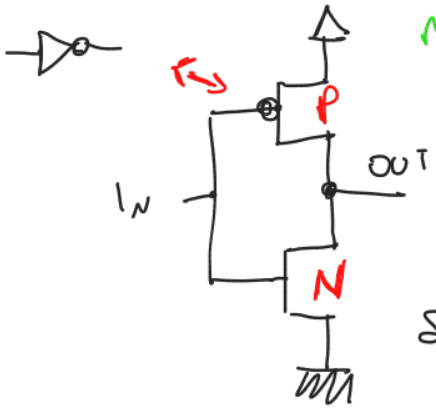


Typical Access Latency (in terms of processor cycles for a 4 GHz processor)

Qureshi+, "Scalable high performance main memory system using phase-change memory technology," ISCA 2009.

Cella di RAM Statica (SRAM)





Scrittura

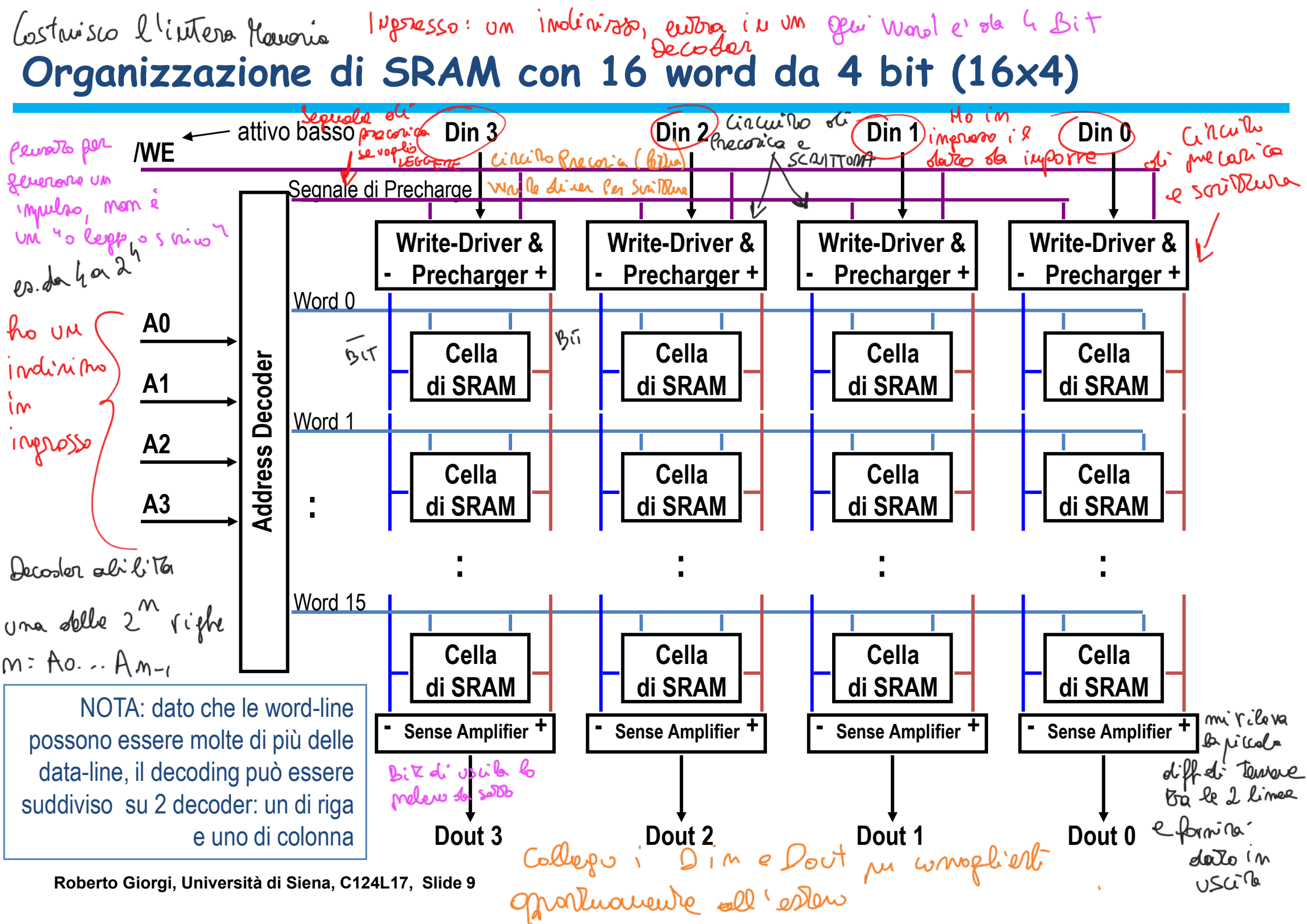
1. Preparare il dato sulle bit-line
2. Selezionare la riga (word-line)

Lettura

1. Precaricare bit e /bit a Vdd *
2. Selezionare la riga (word-line)
3. La cella preleva carica da bit o /bit
4. Il Sense-Amplifier in fondo alla colonna amplifica la differenza fra bit e /bit

*in realtà la linee bit e /bit si caricheranno ad un valore un poco <Vdd

Organizzazione di SRAM con 16 word da 4 bit (16x4)



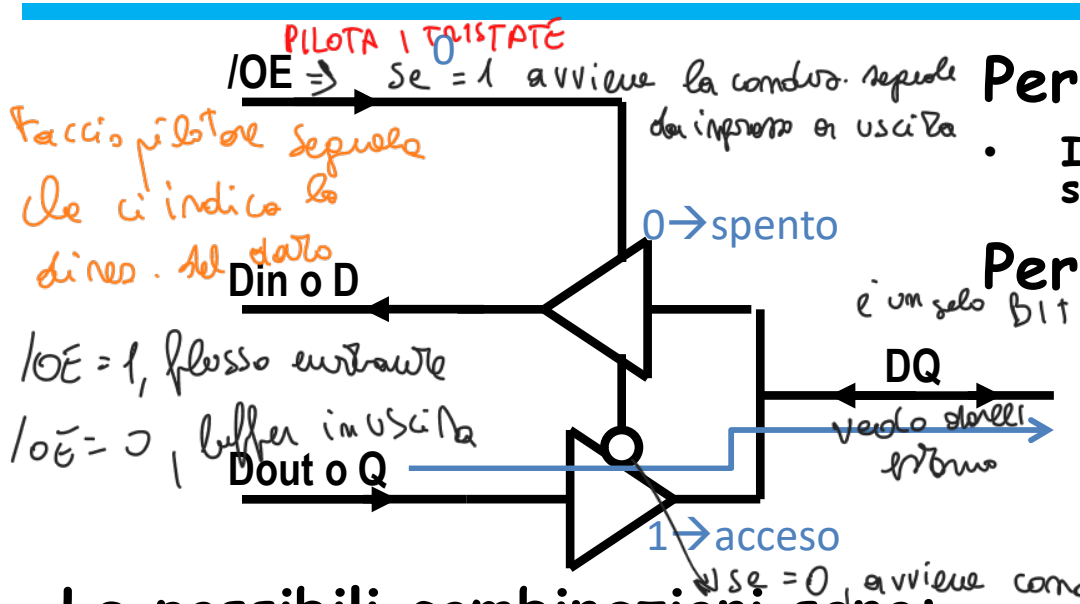
NOTA: dato che le word-line possono essere molte di più delle data-line, il decoding può essere suddiviso su 2 decoder: un di riga e uno di colonna

il Write Enable nei circuiti di memoria è pensato come un impulso nella quale avviene l'operaz. di scrittura

SRAM: Bus-drivers ovvero interfaccia verso il bus dati

collego SRAM al BUS DATI

CIRCUITO con 2 BUFFER TRISTATE



- **Per Din e Dout si usa un unico filo**
  - **Il motivo principale è perché sul bus dati posso avere scambio in entrambe le direzioni (lettura o scrittura)**
- **Per decidere la direzione si usa /OE:**
  - **/OE → Output Enable (attivo basso), per controllare i buffer thristate:**
    1 sul filo tristate → buffer attivo flusso entra
    0 sul file tristate → buffer disattivo

IN	EN	OUT
0	0	Z
1	0	Z
0	1	0
1	1	1

dato di uscita su DQ

- **Le possibili combinazioni sono:**
  - **Scrittura:**
    - **/WE attivo (basso), /OE disattivo (alto) → DQ è un ingresso** (=1)
  - **Lettura:**
    - **/WE disattivo (alto), /OE attivo (basso) → DQ è un'uscita** (=0)
  - **Evitare di attivare contemporaneamente i segnali /WE e /OE**
    - **Il risultato risulta indeterminato**

WE mi dice nel tempo quando avviene effettivamente la scrittura

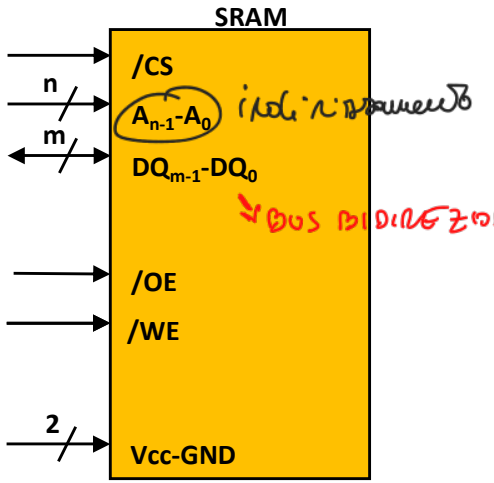
We determina nel tempo il momento

- **Quando il chip non deve caricare il bus dati, si disabilita con /CS**
  - **/CS disattivo → alta impedenza**

stacco il chip

per disabilitare il chip tutti segnali di uscita ad alta impedenza

SRAM: Diagramma Logico



Chip di SRAM da 2^n x m bit

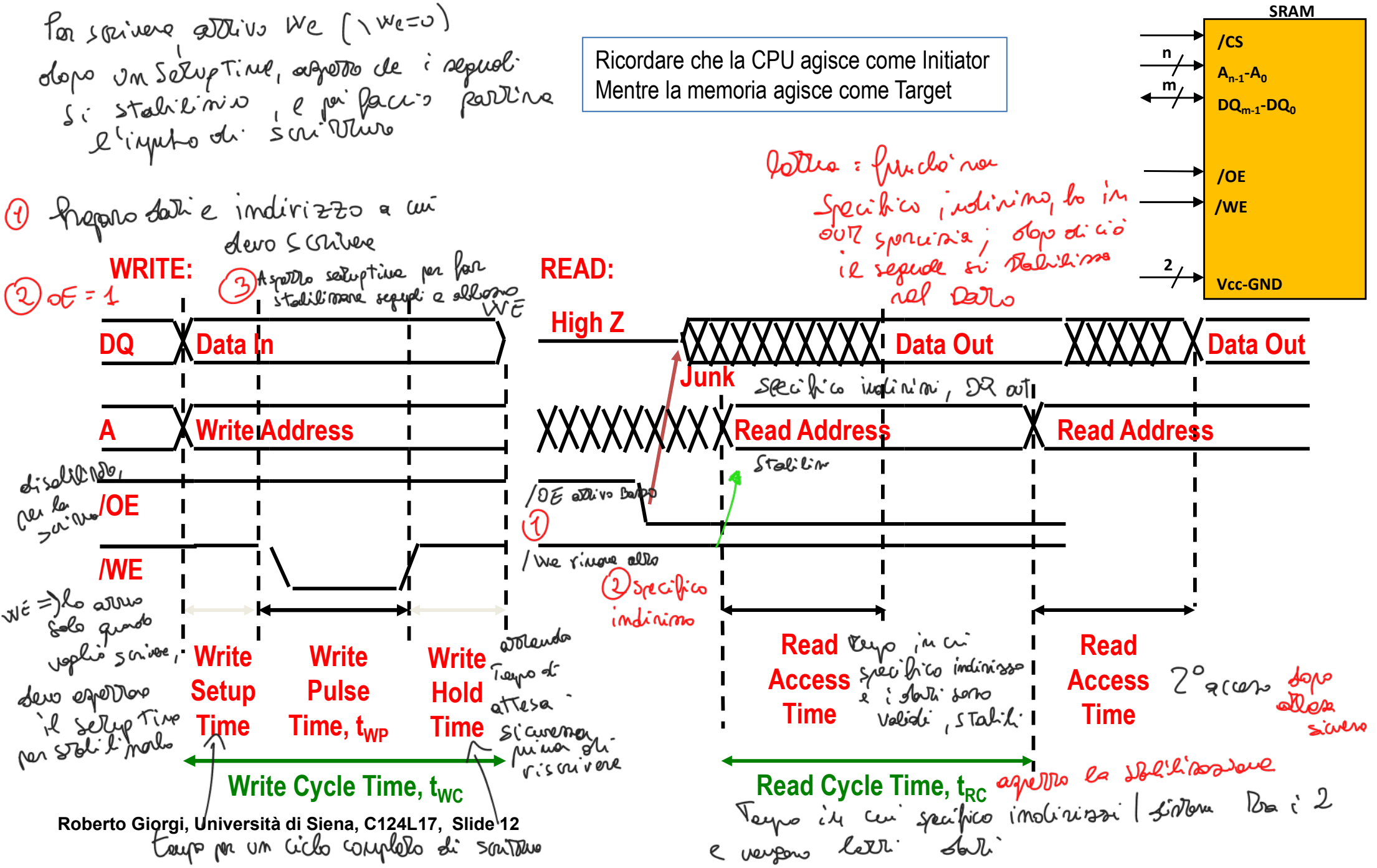
- La memoria ha una capacità di 2^n word ad m-bit

- Altri segnali sempre presenti
  - /CS (o /CE) → Chip Select o Chip Enable
  - Vcc, GND → Alimentazione

SRAM: Temporizzazioni

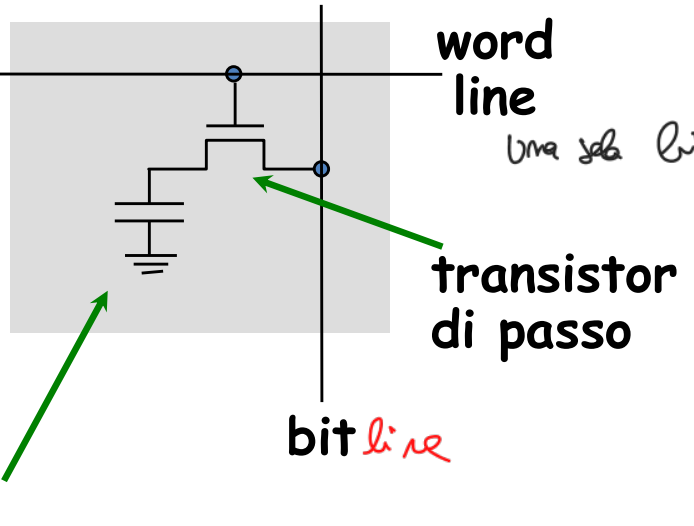
Per scrivere attivo WE (/WE=0)
dopo un Setup Time, appena che i segnali si stabilizzano, e per facci partire l'impulso di scrittura

Ricordare che la CPU agisce come Initiator
Mentre la memoria agisce come Target



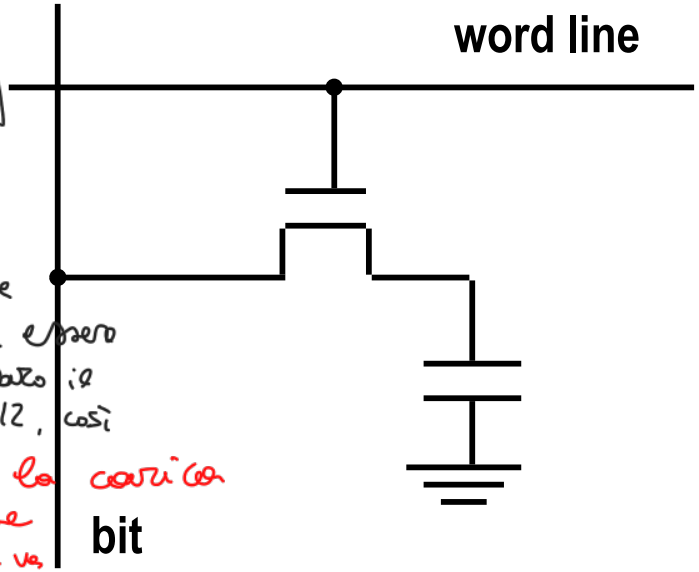
Cella di RAM dinamica (DRAM)

- La cella a 6 transistor per quanto possa sembrare piccola, ha un peso notevole sull'area totale di una RAM
- Qual è il numero minimo di transistor che può essere usato per memorizzare un bit?
- Un solo transistor → l'elemento di memorizzazione può essere realizzato con un condensatore
  - Condensatore carico → 1 logico
  - Condensatore scarico → 0 logico
  - Il transistor consente la lettura e la scrittura



DRAM: Funzionamento della cella a 1 transistor

- **Scrittura:**
  1. Caricare la bit-line col dato da scrivere (0 o 1)
  2. Selezionare la riga (word-line)
- **Lettura:** nel leggere, rileva elettroni del C, devo ripristinare il valore. Inoltre per essere sicuri della dimensione del dato il trucco è precaricare a VDD/2, così se il C era a VDD, la carica va in un verso, se era 0 la carica va nell'altro
  1. Precaricare la bit-line a Vdd/2
  2. Selezionare la riga (word-line)
  3. Scambio di carica fra cella e bit-line
     - lievissima variazione di tensione sulla bit-line
  4. Rilevazione della variazione (Sense-Amplifier molto sofisticato) molto sensibile
     - l'amplificatore deve essere in grado di rilevare una variazione di carica di circa 1 milione di elettroni
  5. Caricare la bit-line col dato letto (scrittura !) lettura = lettura + scrittura
     - Ripristino il contenuto della cella letto il dato, l'ho distrutto
- **Refresh** comunque nel tempo di scarica eventuale il condensatore
  1. Effettuo una "finta" lettura di <u>tutte</u> le celle scarico tutte le celle periodicamente per mantenere il contenuto
     - Ripristino il contenuto di <u>ogni</u> cella il condensatore si scarica

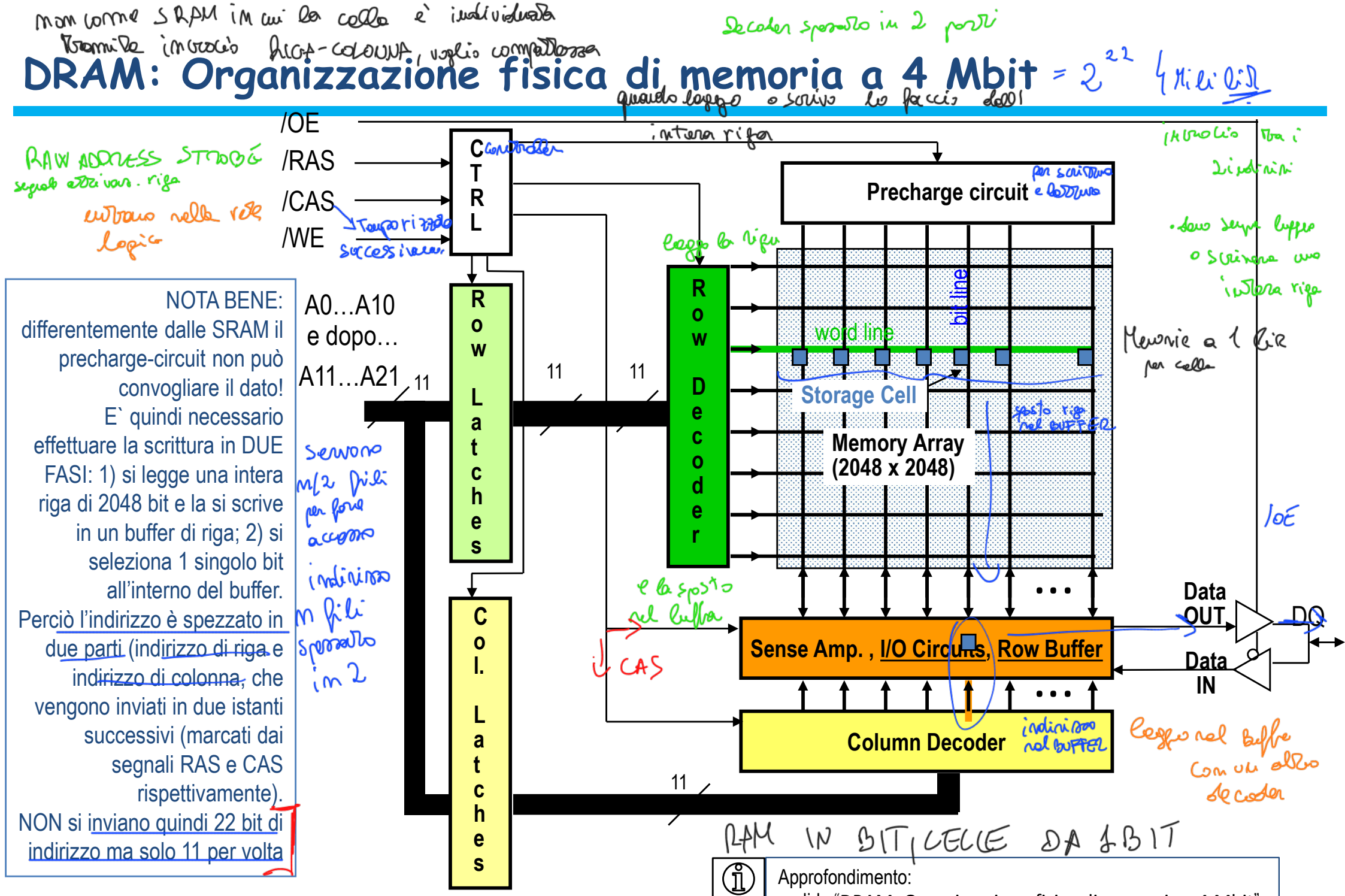


la fa il timer il tempo di Refresh

l'impulso di refresh

N.B. La cella ha <u>un solo filo</u> di accesso (word-line)
→ separazione ROW/COL

DRAM: Organizzazione fisica di memoria a 4 Mbit



NOTA BENE:
differentemente dalle SRAM il precharge-circuit non può convogliare il dato!
E` quindi necessario effettuare la scrittura in DUE FASI: 1) si legge una intera riga di 2048 bit e la si scrive in un buffer di riga; 2) si seleziona 1 singolo bit all'interno del buffer.
Perciò l'indirizzo è spezzato in due parti (indirizzo di riga e indirizzo di colonna, che vengono inviati in due istanti successivi (marcati dai segnali RAS e CAS rispettivamente).
NON si inviano quindi 22 bit di indirizzo ma solo 11 per volta

Approfondimento:
v. slide "DRAM: Organizzazione fisica di memoria a 4 Mbit"

DRAM, più complicata per avere massima compattezza.

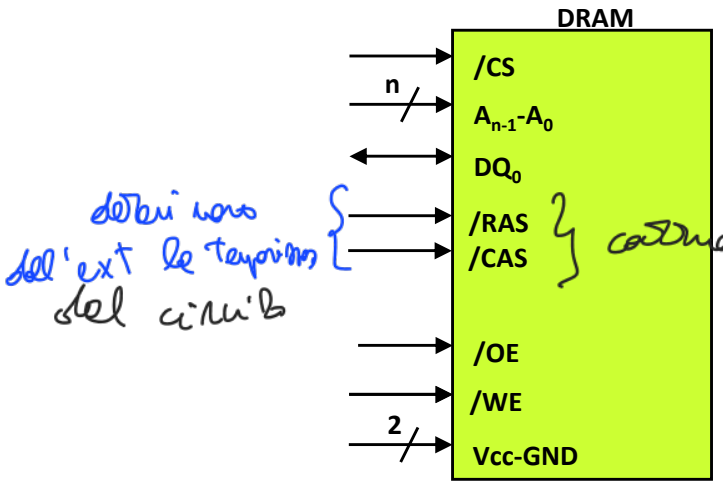
- Decoder deve essere pensato in due parti, una per la parte alta degli indirizzi (decoder di riga), attivato da /RAS, NAW Address STROBE; e altra parte di indirizzi va nel multiplex di colonna

=> Per prelevare il dato devo leggere o scrivere l'intera riga e la riga è composta da quelle che è la Dim. del decoder. Se il decoder indica che ho 2048 linee (2^{11}), ho 11 linee leggo una riga, la sposto nel Buffer, e con l'altro Decoder indirizzo l'interno del Buffer Decoder colonna attivato Grazie CAS, temporizzato

Una volta letto, riscrivo nell'intera riga

4 MBIT, esce un solo bit

DRAM: Visione a livello di chip



Chip di DRAM da 2^{22} x 1 bit

(nella slide precedente n=11 quindi 2^{22} x 1 bit ovvero 4Mbit x1)

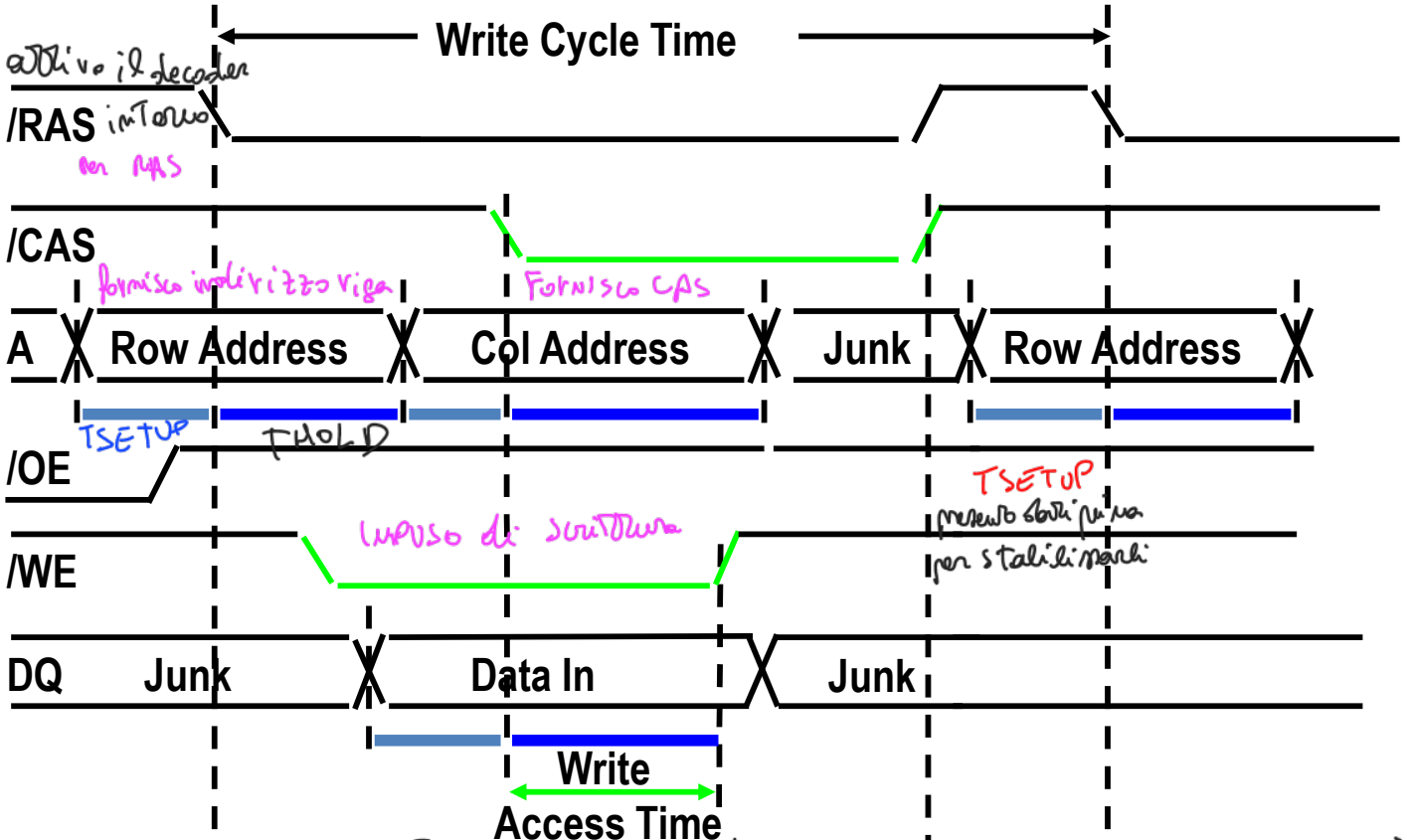
- **Din e Dout sono multiplexati su DQ:**
  - **Scrittura:**
    - /WE attivo (basso), /OE disattivo (alto) → DQ è un ingresso
  - **Lettura:**
    - /WE disattivo (alto), /OE attivo (basso) → DQ è un'uscita
- **Gli indirizzi di Riga e di Colonna condividono i pin A**
  - /RAS va basso → i latch di riga memorizzano i pin A
  - /CAS va basso → i latch di colonna memorizzano i pin A
  - Nota: RAS/CAS sono sensibili al fronte in discesa

Nota: In alcuni chip di DRAM il segnale OE non c'è e invece di DQ ci sono Din e Dout

DRAM: Ciclo di scrittura

- L'accesso inizia con l'attivazione di /RAS

fornisco l'indirizzo specificando prima gli 11 bit di riga, attivando RAS; aspetto THold, poi metto indirizzo CAS



Tempo da quando CAS è attivo e l'impulso di scrittura termina

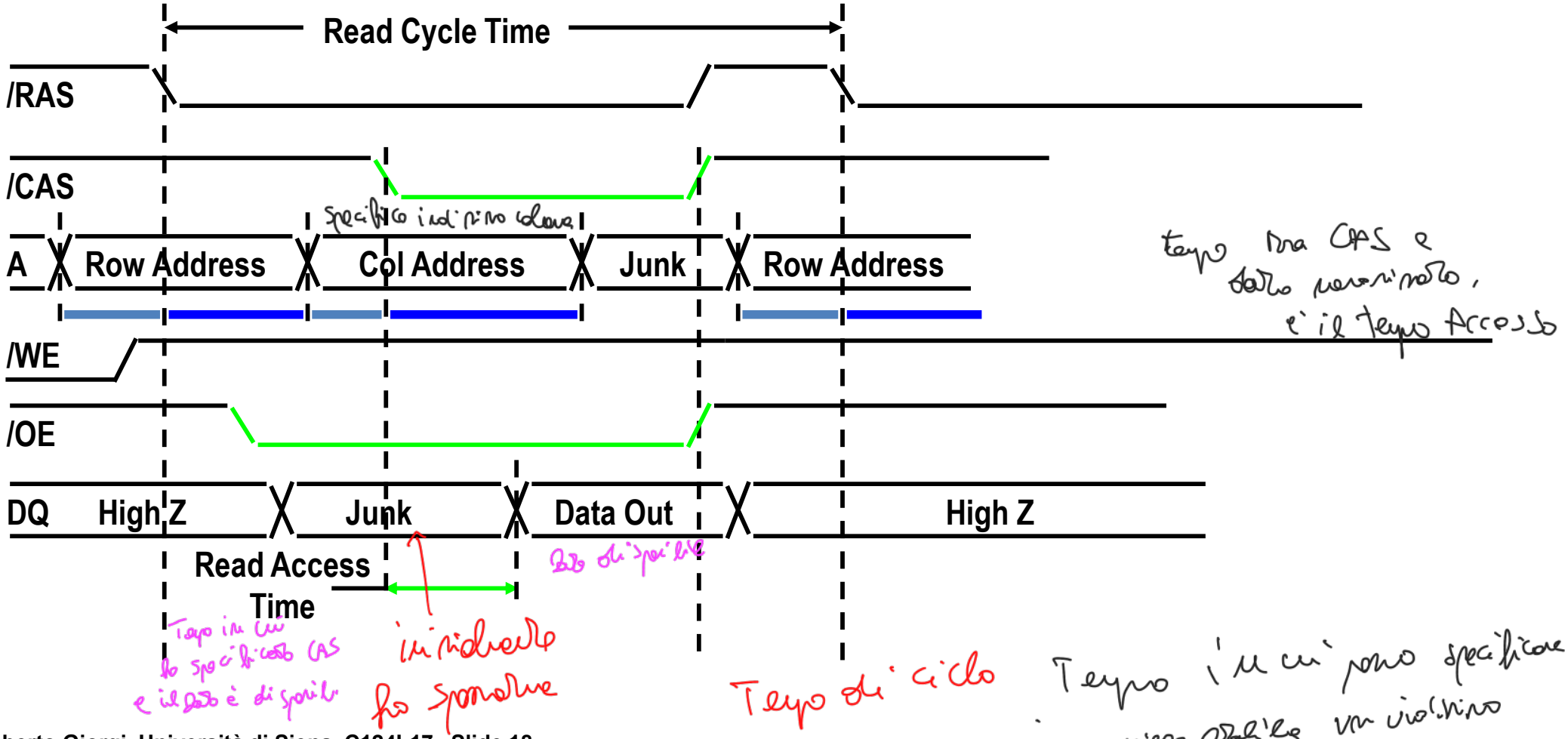
Tutto scritto quando c'è il segnale di CAS

- Per la scrittura deve servire il segnale di CAS.

T_{WRITE} = Tempo da quando ho attivato CAS a quando termina l'IMPULSO WE

DRAM: Ciclo di lettura

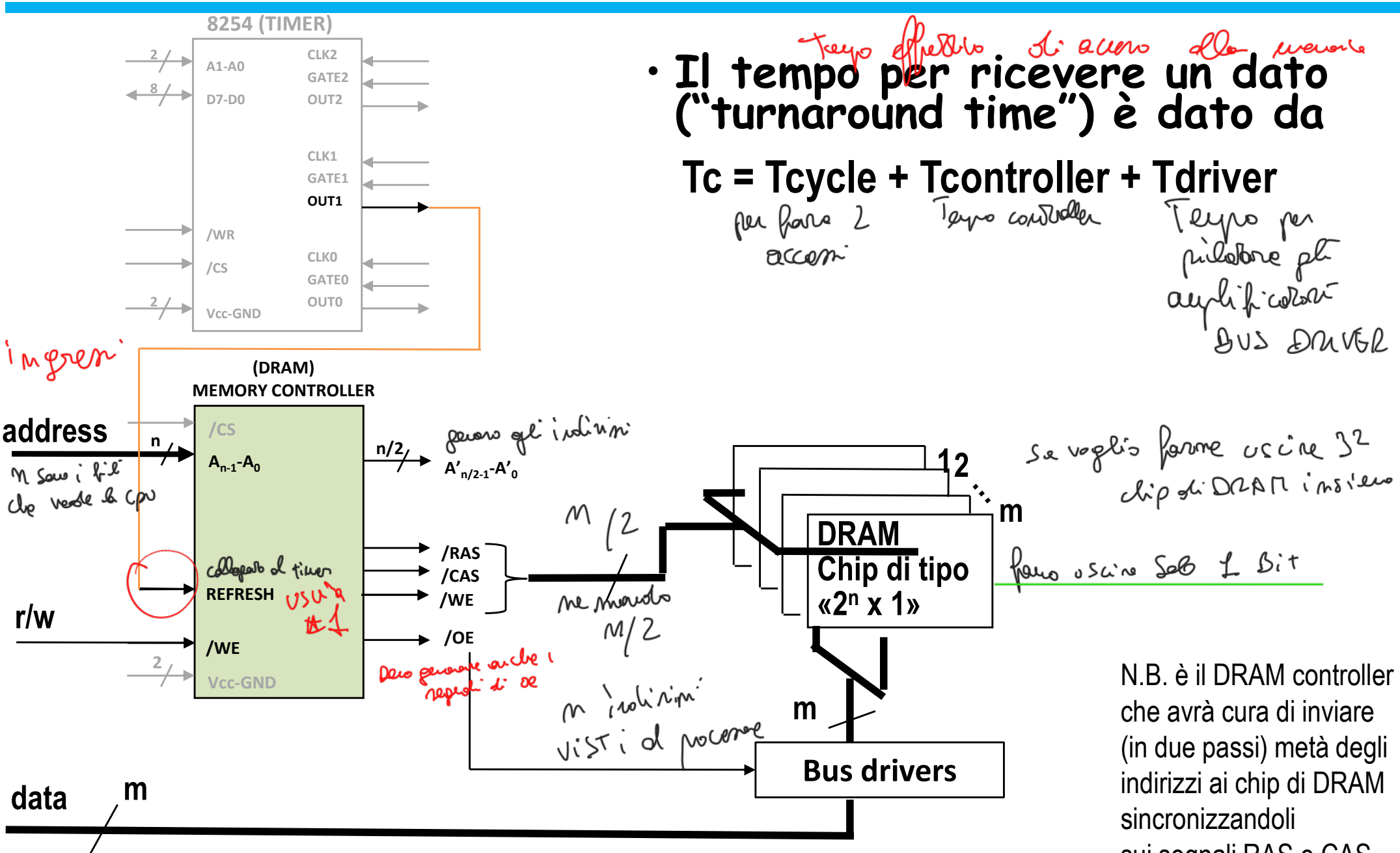
- L'accesso inizia con l'attivazione di /RAS



DRAM Controller (Memory Controller)

- Il tempo per ricevere un dato ("turnaround time") è dato da

$$Tc = Tcycle + Tcontroller + Tdriver$$



N.B. è il DRAM controller che avrà cura di inviare (in due passi) metà degli indirizzi ai chip di DRAM sincronizzandoli sui segnali RAS e CAS